DSI'05, Decision Sciences Institute International Conference
Barcelona (Spain), 3-6 July 2005, pp. 159-169.

Modelling and Forecasting financial time series of «tick data» by functional analysis and neural networks

Simon DABLEMONT, Michel VERLEYSEN

Université catholique de Louvain, Machine Learning Group, DICE
3, Place du Levant, B-1348 Louvain-la-Neuve - BELGIUM
Tel : +32 10 47 25 51 - Fax : +32 10 47 25 98
E-mail : {dablemont, verleysen}@dice.ucl.ac.be

Abstract

A functional method for time series forecasting is presented. Based on the splitting of the past dynamics into clusters, local models are built to capture the possible evolution of the series given the last known values. A probabilistic model is used to combine the local predictions. The method can be applied to any time series forecasting problem, but is particularly suited to data showing nonlinear dependencies, cluster effects, and observed at irregularly and randomly spaced times as financial series of "tick data" do. The method is applied to the forecasting of financial time series of «tick data» of IBM asset.

Keywords

Functional, Sparse curves, Clustering, "tick data", Forecasting, Neural Networks

1 Introduction

The analysis of financial time series is of primary importance in the economical world. This paper deals with a data-driven empirical analysis of financial time series; the goal is to obtain insights into the dynamic of the series and out-of-sample forecasting.
Forecasting future returns on assets is of obvious interest in empirical finance. If one were able to forecast tomorrow's return on an asset with some degree of precision, one could use this information in an investment today. Unfortunately, we are seldom able to generate a very accurate prediction for asset returns.
Financial time series display typical nonlinear characteristics : it exists clusters within which returns and volatility display specific dynamic behavior. For this reason, we will consider here nonlinear forecasting models, based on local analysis into clusters. Although financial theory does not provide many motivations for nonlinear models, analyzing data by nonlinear tools seems to be appropriate, and is at least as much informative as an analysis by more restrictive linear methods.
Time series of asset returns can be characterized as serial dependent. This is revealed by the presence of positive autocorrelation in squared returns, and sometimes in the returns

DSI'05, Decision Sciences Institute International Conference
Barcelona (Spain), 3-6 July 2005, pp. 159-169.

too. The increased importance played by risk and uncertainty considerations in modern economic theory, has necessitated the development of new econometric time series techniques that allow for modelling of time varying means, variances and covariances. Given the apparent lack of any structural dynamic economic theory explaining the variation in the second moment, econometricians have thus extended traditional time series tools such as AutoRegressive Moving Average (ARMA) models (Box and Jenkins 1970) for the conditional means and equivalent models for the conditional variance. Indeed, the dynamics observed in the dispersion is clearly the dominating feature in the data. The most widespread modelling approach to capture these properties is to specify a dynamic model for the conditional mean and the conditional variance, such as an ARMA-GARCH model or one of its various extensions (Engle 1982), (Hamilton 1994).

The Gaussian random walk paradigm - under the form of the diffusion geometric Wiener process - is the core of modelling of financial time series. Its robustness mostly suffices to keep it as the best foundation for any development in financial modelling, in addition to the fact that, on the long run, and with enough spaced out data, it is almost verified by the facts. Failures in its application are however well admitted on the (very) short term (market microstructure) (Fama 1991). We claim that, to some extent, such failures are actually caused by the uniqueness of the modelling process.

The first breach in such unique process has appeared with two-regime or switching processes (Diebold 1994), which recognize that a return process could be originated by two different stochastic differential equations. But in such case, the switch is governed by an exogenous cause (for example in the case of exchange rates, the occurrence of a central bank decision to modify its leading interest rate or to organize a huge buying or selling of its currency through major banks) .

Market practitioners, however, have always observed that financial markets can follow different behaviors over time, such as overreaction, mean reversion, etc, which look like succeeding each other with the passing time. Such observations would justify a rather fundamental divergence from the classic modelling foundations. That is, financial markets should not be modelled by a single process, but rather by a succession of different processes, even in absence of the exogenous causes retained by existing switching process. Such a multiple switching process should imply, first, the determination of a limited number of competitive sub-process, and secondly, the identification of the factor(s) causing the switch from one to another sub-processes. The resulting model should not be Markovien, and, without doubt, would be hard to determine. The aim of this paper is, as a first step, to at least empirically verify, with the help of functional clustering and neural networks, that a multiple switching process leads to better short term forecasting.

In this paper we will present a forecasting method based on an empirical functional analysis of the past of the series. An originality of this method is that it does not make the assumption that a single model is able to capture the dynamics of the whole series. On the contrary, it splits the past of the series into clusters, and generates a specific local neural model for each of them. The local models are then combined in a probabilistic way, according to the distribution of the series in the past. This forecasting method can be applied to any time series forecasting problem, but is particularly suited for data showing nonlinear dependencies, cluster effects and observed at irregularly and randomly spaced times like financial time series of «tick data» do. One way to overcome the irregular and random sampling of "tick-data" is to resample them at low frequency, as it is done with "Intraday". However, even with optimal resampling using say five minute returns when transactions are recorded every second, a vast amount of data is discarded, in contradiction to basic statistical principles. Thus modelling the noise and using all the data is a

DSI05, Decision Sciences Institute International Conference
Barcelona (Spain), 3-6 July 2005, pp. 159-169.
better solution, even if one misspecifies the noise distribution (Aït-Sahalia 2003).
In the following of this paper, we will first describe how Functional Analysis can be applied to time series data (section 2), and briefly introduce the Radial-Basis Functions Networks that will be used as nonlinear models (section 3). Then, we will describe the forecasting method itself (section 4), and illustrate its results on the IBM series of "tick data" (section 5).

2 Functional Modelling and Clustering

Cluster analysis consists in identifying groups in data; it is the dual form of discriminant analysis but in cluster analysis the group labels are not known a priori. We assume that the observations $[\mathbf{y}_1, \cdots, \mathbf{y}_N]$ are generated according to a mixture distribution with G components. Let $f_k(\mathbf{y}|\theta_k)$ be the density corresponding to cluster k, with parameters θ_k, and let $1_{\{k\}}(i)$ be the cluster membership (indicator function of cluster k) for the observation i where $1_{\{k\}}(i) = 1$ if y_i is a member of cluster k and 0 otherwise. The indicators are unknown and $1_{\{k\}}(i)$ is multinomial with parameters $[\pi_1, \cdots, \pi_G]$ and π_k is the probability that an observation belongs to cluster k. We can estimate the parameters by maximizing the likelihood

$$L(\theta_1, \cdots, \theta_G; \pi_1, \cdots, \pi_G | \mathbf{y}_1, \cdots, \mathbf{y}_N) = \prod_{i=1}^{N} \sum_{k=1}^{G} \pi_k f_k(\mathbf{y}_i|\theta_k). \tag{1}$$

The maximum likelihood corresponds to the most probable model, given the observations $\mathbf{y}_1, \cdots, \mathbf{y}_N$. Such model can be used in finite dimensional problems, but it is not appropriated to infinite dimensional data such as curves. We could run around by discretizing the time interval, but generally the resulting data vectors are highly correlated and high-dimensional, and by resampling at low frequency we loose much information. Another approach is to projet each curve onto a finite-dimensional basis $\phi(x)$, and find the best projection of each curve onto this basis. The resulting basis coefficients can than be used as a finite-dimensional representation making it possible to use classical clustering methods on the basis coefficients. These approches can work well when every curve has been observed over the same fine grid of points, but they break down if the individual curves are sparsely sampled. In this case, we convert the original infinite dimensional problem into a finite dimensional one using basic functions and we use a random effects model for the coefficients (Rice 2001).

2.1 Modelling functional data

Let $\mathbf{h}_i$, $\mathbf{Y}_i$ and ϵ_i be the vectors of the true value, observed values and measurement errors for the curve i, at times $(t_{i1}, \cdots, t_{in_i})$. Remember that the curves are irregularly and differently sampled, so that the number of observation in curve i depends on i, it is noted n_i. Then

$$\mathbf{Y}_i = \mathbf{h}_i + \epsilon_i, \quad \text{for} \qquad i = 1, \cdots, N, \tag{2}$$

where N is the number of curves. The measurement errors are assumed to have zero mean and to be uncorrelated with each other and with $\mathbf{h}_i$. We chose natural cubic splines as basic functions because they have desirable mathematical properties, are easy to implement and require a relative minimal number of parametric assumptions (deBoor

(2005) Decision Sciences Institute International Conference
Barcelona (Spain), 3-6 July 2005, pp. 159-169.
The functional clustering model can be written as

$$\mathbf{Y}_i = \mathbf{S}_i(m + \mathbf{A}\mathbf{c}_{\mathbf{z}_i} + \mathbf{d}_i) + \epsilon_i, \quad \text{for} \quad i = 1, \cdots, N, \tag{3}$$

where

- $\epsilon_i \sim N(\mathbf{0}, R)$,
- $d_i \sim N(\mathbf{0}, Q)$,
- $\mathbf{R} = \sigma^2\mathbf{I}$,
- $\mathbf{Q}$ is the same for all clusters,
- $\mathbf{S}_i = (\mathbf{s}_i(t_{i1}), \cdots, \mathbf{s}_i(t_{in_i}))^T$ is the spline basis matrix for curve i,
- $\mathbf{m}$ is a vector of dimension $[q]$,
- $\mathbf{c}_k$ is a vector of dimension $[h]$,
- $\mathbf{A}$ is a matrix $[q, h]$, with $h \leq \min(q, G-1)$,
- G is the number of clusters,
- $\mathbf{z}_i$ denotes the unknown cluster membership vector for curve i, with $z_{ik} = 1$ if curve i is a member of cluster k and 0 otherwise,
- $\mathbf{s}(t)^T\mathbf{m}$: modelling of the mean curve,
- $\mathbf{s}(t)^T\mathbf{A}\mathbf{c}_k$: variation of the centroid with respect to the mean curve,
- $\mathbf{s}(t)^T\mathbf{d}_i$: variation of the curve with respect to the centroid of its cluster,
- $\mathbf{s}(t)$ is a spline basis vector of dimension $[q]$. Let us for example choose a cubic power spline with 3 nodes; we then have $\mathbf{s}(t) = \big[s_1(t), \cdots, s_5(t)\big]^T$, with $s_1(t) = 1, s_2(t) = t, s_3(t) = t^2, s_4(t) = t^3, s_5(t) = (t-\tau)^3_+$, where τ is the location of the internal node.

2.2 The fitting algorithm

We need to identify the parameters $\mathbf{m}$, $\mathbf{A}$, $\mathbf{c_k}$, $\mathbf{Q}$, σ^2 et π_k by maximisation of the likelihood. The likelihood fitting procedure treats the unknown cluster membership $\mathbf{z}_i$ as missing data and uses the EM algorithm (Dempster 1977). Note that since the $\mathbf{z}_i$ and the d_i are assumed independent one from another, the complete data distribution factors as

$$f(\mathbf{Y}, \mathbf{z}_i, d) = f(\mathbf{Y}|\mathbf{z}_i, d) f(\mathbf{z}_i) f(d). \tag{4}$$

Given that the $\mathbf{z}_i$ are multinomial with parameters π_k, the d_i are $N(0, \mathbf{Q})$ and the $\mathbf{Y}_i$ are conditional $N[S_i(\mathbf{m} + \mathbf{A}\mathbf{c}_k + \mathbf{d}_i), \sigma^2\mathbf{I}]$ the complete data likelihood is

$$L(\pi_k, \mathbf{m}, \mathbf{A}, \mathbf{c}_k, \mathbf{Q}, \sigma^2 | \mathbf{Y}_i, \mathbf{z}_i, \mathbf{d}_i) = \prod_{i=1}^{N} (2\pi)^{-\frac{n_i+q}{2}} |\mathbf{Q}|^{-\frac{1}{2}} \exp\big(-\frac{1}{2}\mathbf{d}_i^T\mathbf{Q}^{-1}\mathbf{d}_i\big) *$$
$$\sum_{k=1}^{G} \Big\{ \pi_k \exp(-\frac{n_i}{2}\log\sigma^2) \exp\big(-\frac{1}{2\sigma^2}\big\|\mathbf{Y}_i - \mathbf{S}_i(\mathbf{m} + \mathbf{A}\mathbf{c}_k + \mathbf{d}_i)\big\|^2\big) \Big\}^{z_{ik}}. \tag{5}$$

DSI'05 Decision Sciences Institute International Conference
Barcelona (Spain), 3-6 July 2005, pp. 159-169.

The EM algorithm consists of iteratively maximizing the expected values of (5) given $\mathbf{Y}_i$ and the current parameters estimates. Since all three parts of the complete data log likelihood of (5) involve separate parameters they can be maximized independently of each other.

2.3 E-step

The E-step consists of predicting $\widehat{\mathbf{d}}_i = E\Big\{\mathbf{d}_i|\mathbf{Y}_i, \mathbf{m}, \mathbf{A}, \mathbf{c}, \mathbf{Q}, \sigma^2, \mathbf{z}_i\Big\}$. The result is given by

$$\widehat{\mathbf{d}_\mathbf{i}} = \big(\mathbf{S}_i^T\mathbf{S}_i + \sigma^2\mathbf{Q}^{-1}\big)^{-1}\mathbf{S}_i^T\big(\mathbf{Y}_i - \mathbf{S}_i m - \mathbf{S}_i\mathbf{A}c_k\big). \tag{6}$$

2.4 M-step

The M-step maximizes $Q = E\{l(\pi_k, \mathbf{m}, \mathbf{A}, \mathbf{c}_k, \mathbf{Q}, \sigma^2|\mathbf{Y}_i, \mathbf{z}_i, \mathbf{d}_i)\}$.

2.4.1 Estimation of π_k

We have

$$\widehat{\pi}_k = \frac{1}{N}\sum_{i=1}^{N}\pi_{k|i}. \tag{7}$$

with $\pi_{k|i} = P\big(z_{ik} = 1|\mathbf{Y}_i\big) = \frac{f(y|z_{ik})\pi_k}{\sum_{j=1}^{G} f(y|z_{ik})\pi_j}$,
and $f(y|z_{ik})$ is given by $\mathbf{Y}_i \sim N(\mathbf{S}_i(\mathbf{m} + \mathbf{A}\mathbf{c}_{\mathbf{z}_i}), \mathbf{\Sigma}_i)$, where $\mathbf{\Sigma}_i = \sigma^2\mathbf{I} + \mathbf{S}_i\mathbf{Q}\mathbf{S}_i^T$.

2.4.2 Estimation of Q

We have

$$\widehat{\mathbf{Q}} = \frac{1}{N}\sum_{i=1}^{N} E\big[\widehat{\mathbf{d}}_i\widehat{\mathbf{d}}_i^T|\mathbf{Y}_i\big] = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{G} E\big[\widehat{\mathbf{d}}_i\widehat{\mathbf{d}}_i^T|\mathbf{Y}_i, z_{ik} = 1\big]. \tag{8}$$

with $(\widehat{\mathbf{d}}_i|\mathbf{Y}_i, z_{ik})$ given by the E-step.

2.4.3 Estimation of m, $\mathbf{c}_k$ and A

This involve an iterative procedure where $\mathbf{m}$, $\mathbf{c}_k$ and the columns of $\mathbf{A}$ are repeatedly optimized while holding all other parameters fixed.

2.4.4 Estimation of m

First let $\mathbf{Y}_i = \mathbf{S}_i(\mathbf{m} + \mathbf{A}\mathbf{c}_{\mathbf{z}_i} + \mathbf{d}_i) + \epsilon_i, \qquad \text{for} \quad i = 1, \cdots, N.$
By GLS we get

$$\widehat{\mathbf{m}} = \Big(\sum_{i=1}^{N}\mathbf{S}_i^T\mathbf{S}_i\Big)^{-1}\sum_{i=1}^{N}\mathbf{S}_i^T\Big[\mathbf{Y}_i - \sum_{k=1}^{G}\pi_{k|i}\mathbf{S}_i(\widehat{\mathbf{A}}\widehat{\mathbf{c}}_k + \widehat{d}_{ik})\Big], \tag{9}$$

with $\widehat{d}_{ik} = E\big\{d_{ik}|z_{ik}, \mathbf{Y}_i\big\}$ given by the E-step.

2.4.5 Estimation of $\mathbf{c}_k$

The $\widehat{\mathbf{c}}_k$ are calculated using

$$\widehat{\mathbf{c}}_k = \Big(\sum_{i=1}^{N} \pi_{k|i} \widehat{\mathbf{A}}^T \mathbf{S}_i^T \mathbf{S}_i \widehat{\mathbf{A}} \Big)^{-1} \sum_{i=1}^{N} \pi_{k|i} \widehat{\mathbf{A}}^T \mathbf{S}_i^T \Big[\mathbf{Y}_i - \mathbf{S}_i \widehat{\mathbf{m}} - \mathbf{S}_i \widehat{d}_{ik} \Big]. \tag{10}$$

2.4.6 Estimation of A

By GLS we only compute vectors, thus each column of $\mathbf{A}$ is optimized holding all other fixed using

$$\widehat{\mathbf{a}}_m = \Big(\sum_{i=1}^{N} \sum_{k=1}^{G} \pi_{k|i} \widehat{c}_{km}^2 \mathbf{S}_i^T \mathbf{S}_i \Big)^{-1} \sum_{i=1}^{N} \sum_{k=1}^{G} \pi_{k|i} \widehat{c}_{km} \mathbf{S}_i^T \Big(\mathbf{Y}_i - \mathbf{S}_i \widehat{\mathbf{m}} - \sum_{l \neq m}^{G} \widehat{c}_{km} \mathbf{S}_i \widehat{\mathbf{a}}_l - \mathbf{S}_i \widehat{d}_{ik} \Big), \tag{11}$$

where

- $\mathbf{a}_m$ is the mth column of $\mathbf{A}$
- $\widehat{c}_{km}$ is the mth component of $\widehat{c}_k$

We iterates through (9), (10) and (11) until all parameters have converged.

2.4.7 Estimation of σ^2

The final step is to set

$$\widehat{\sigma^2} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{G} \widehat{\pi}_k \Big\{ \overline{\mathbf{y}}_{ik}^T \overline{\mathbf{y}}_{ik} + \mathbf{S}_i Cov[\widehat{\mathbf{d}}_i | \mathbf{Y}_i, z_{ik}] \mathbf{S}_i^T \Big\}, \tag{12}$$

where $\overline{\mathbf{y}}_{ik} = \mathbf{Y}_i - \mathbf{S}_i \widehat{\mathbf{m}} - \mathbf{S}_i \widehat{\mathbf{A}} \widehat{\mathbf{c}}_k - \mathbf{S}_i \widehat{\mathbf{d}}_i$

The algorithm iterates through (7), (8), (9), (10), (11) and (12) until all parameters have converged.

3 Radial Basis Function Networks

Radial Basis Function Networks (RBFN) are neural networks used in approximation and classification tasks. They share with Multi-Layer Perceptrons the universal approximation property (Haykin 1999).
Classical RBF networks have their inputs fully connected to non-linear units in a single hidden layer. The output of a RBFN is a linear combination of the hidden units outputs. More precisely, the output is a weighted sum of Gaussian functions or kernels (i.e. the nonlinearities) applied to the inputs :

$$y \;=\; \sum_{i=1}^{I} \lambda_i \exp \Big\{ - \frac{|| \; x - c_i \; ||^2}{\sigma_i} \Big\}. \tag{13}$$

where x is the input vector, y is the scalar output of the RBFN, $c_i,\ 1 \leq i \leq I$, are the centers of the I Gaussian kernels, $\sigma_i,\ 1 \leq i \leq I$, are their widths, and $\lambda_i,\ 1 \leq i \leq I$,

DSI'05, Decision Sciences Institute International Conference
Barcelona (Spain), 3-6 July 2005, pp. 159-169.
their weights; intuitively those last λ_i parameters represent the relative importance of each kernel in the output y.

As shown in equation (13), the RBF network has three sets of parameters $c_i,\ \sigma_i, \lambda_i, 1 \leq i \leq I$. One advantage of RBFN networks compared to other approximation models is that these three sets can be learned separately with suitable performances. Moreover the learning of the λ_i weights results from a linear system. A description of learning algorithms for RBF networks can be found in (Verleysen 2003).

4 The Forecasting Method

In this section we present a detailed model-based approach for clustering functional data and a time series forecasting method. This method will first be sketched to give an intuition of how the forecasting is performed. Then each step of the method will be detailed.

4.1 Method Description

The forecasting method is based on the "looking in the past" principle. To perform a functional prediction of the curve for the time interval $[t, t + \triangle t_{out}]$, we create two functional spaces. A first functional space IN is built with past observations for the time interval $[t - \triangle t_{in}, t]$, the "regressors" and a similar second functional space OUT is built with observations for the time interval $[t - \triangle t_{in}, t + \triangle t_{out}]$; these two spaces are built with all data corresponding to times $t \in [t_0 + \triangle t_{in}, T - \triangle t_{in} - \triangle t_{out}]$. These functional spaces are combined into a probabilistic way to build the functional prediction for the time interval $[t, t + \triangle t_{out}]$ and are quantized using the functional clustering algorithm. The relationship between the first and the second functional spaces issued from the clustering algorithms is encoded into a frequency table constructed empirically on the datasets. In each of the clusters determined by the second clustering OUT, a local RBFN model is built to approximate the relationship between the functional output (the local prediction) and the functional input (the regressor).
Finally, the global functional prediction at time t for the interval $[t, t+\triangle t_{out}]$ is performed by combining the local models results associated to clusters OUT, according to their frequencies with respect to the class considered in the cluster IN.

4.2 Quantizing the « inputs »

Consider a scalar time series X, where $x(t)$ is the value at time t, $t \in [t_0, T]$. This original series is transformed into an array of observations X_{in} for the time intervals $[t, t + \triangle t_{in}]$, for all $t \in [t_0, t_0 + \triangle t_{in}, t_0 + 2\triangle t_{in}, \cdots, T - \triangle t_{in} - \triangle t_{out}]$. Then the clustering algorithm is applied to the *input* array X_{in}; after convergence it gives an IN map of K_{in} codewords and the spline coefficients for the curves of each cluster in this IN map .

4.3 Quantizing the « outputs »

At each input vector of the matrix X_{in} we aggregate the next observations to get a new array Y_{out} for the time interval $[t, t+\triangle t_{in}+\triangle t_{out}]$ for all $t \in [t_0, t_0+\triangle t_{in}, t_0+2\triangle t_{in}, \cdots, T-\triangle t_{in}-\triangle t_{out}]$. The clustering algorithm is applied to the new array Y_{out}; after convergence it gives an OUT map of K_{out} codewords and the spline coefficients for the curves of each

DSI'05 Decision Sciences Institute International Conference
Barcelona (Spain), 3-6 July 2005, pp. 159-169.
cluster in this OUT map. Note that, by construction, there is a one-to-one relationship between each *input* and each *output* vector of spline coefficients.

4.4 Frequency table

The two sets of codewords from maps IN and OUT only contain a static information. This information does not reflect completely the evolution of the time series. The idea is thus to create a data structure that represents the dynamics of the time series, i.e. how each class of *output* vectors of spline coefficients (including the values for the time interval $[t, t + \triangle t_{out}]$) is associated to each class of *input* vectors of spline coefficients for the time interval $[t - \triangle t_{in}, t]$. This structure is the frequency table $T(i, j)$, with $1 \leq i \leq N_{in}$, $1 \leq j \leq N_{out}$. Each element $T(i, j)$ of this table represents the proportion of *output* vectors that belongs to the j^{th} class of the OUT map while their corresponding *input* vectors belong to class i of the IN map. Those proportions are computed empirically for the given dataset and sum to one on each line of the table. Intuitively the frequency table represents all the possible evolutions at a given time t together with the probability that they effectively happen.

4.5 Local RBFN models

When applied to the « outputs », the functional clustering algorithm provides N_{out} classes and the spline coefficients of the curves for the intervals $[t, t + \triangle t_{out}]$. In each of these classes a RBFN model is learned. Each RBFN model has q inputs (the spline's coefficients of the regressors) and q outputs (the spline coefficients of the prediction curve). These models represent the local evolution of the time series, restricted to a specific class of regressors. The local information provided by these models will be used when predicting the future evolution of the time series.

4.6 Forecasting

The relevant information has been extracted from the time series through both maps, the frequency table and the local RBFN models detailed in the previous sections. Having this information, it is now possible to perform the forecasting itself. At each time t, the goal is to estimate the functional curve for the time interval $[t, t+\triangle t_{out}]$ denoted $\hat{\mathbf{x}}([t, t+\triangle t_{out}])$. First the *input* at time t is built, leading to $X(t)$. This vector is presented to the IN map, and the nearest codeword $X_{k(t)}$ is identified ($1 \leq k(t) \leq N_{in}$).
In the frequency table, in the $k(t)^{\text{th}}$ line, there are some columns corresponding to classes of the OUT map for which the proportions are non zero. This means that those columns represent possible evolutions for the considered data $X(t)$, since $X(t)$ has the same shape than data in the $k(t)^{\text{th}}$ class.

For each of those potential evolutions, the respective RBFN models are considered (one RBFN model has been built for each class in the OUT map). For each of them, a local prediction $\hat{x}_j([t, t + \triangle t_{out}])$ is obtained ($1 \leq j \leq N_{out}$). The final prediction is a weighted sum of the different local predictions, the weights being the proportions recorded in the frequency table. The final prediction is thus

$$\hat{\mathbf{x}}([t, t + \triangle t_{out}]) = \sum_{j=1}^{N_{out}} T(k, j)\hat{\mathbf{x}}_j([t, t + \triangle t_{out}]). \quad (14)$$

DSI'05, Decision Sciences Institute International Conference
Barcelona (Spain), 3-6 July 2005, pp. 159-169.

5 Experimental Results

The examples presented here deal with the IBM stock time series of "tick data" for the period starting on January 02, 1997 and ending on may 08, 1997 with more than 3000 transactions per day, on the New York Stock Exchange (NYSE).
On Fig. 1 we can see the evolution of the Prices (top) and Volumes (bottom) on one day. On Fig. 2 we see the distribution of transactions for the same day. Each point is a transaction, with more transactions at the opening and closing of the NYSE. The transactions are sampled discretely in time and like it is often the case with financial data the time separating successive observations is itself random.

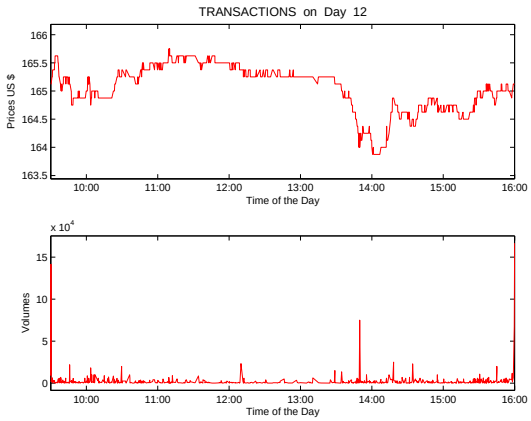

Figure 1: Prices (Top) et Volumes (Bottom)

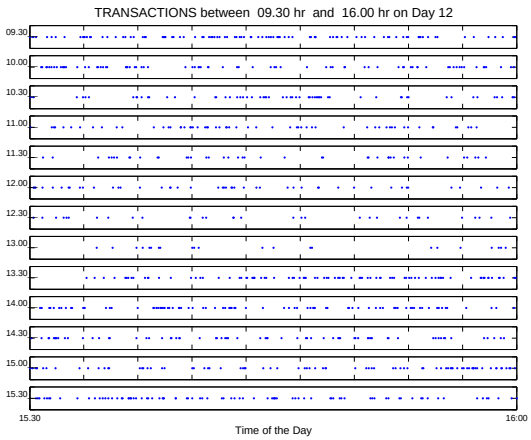

Figure 2: Distributions of transactions

On Fig. 3 we can see two successives days of the stock IBM with a fine smoothing of the "tick data" by splines.

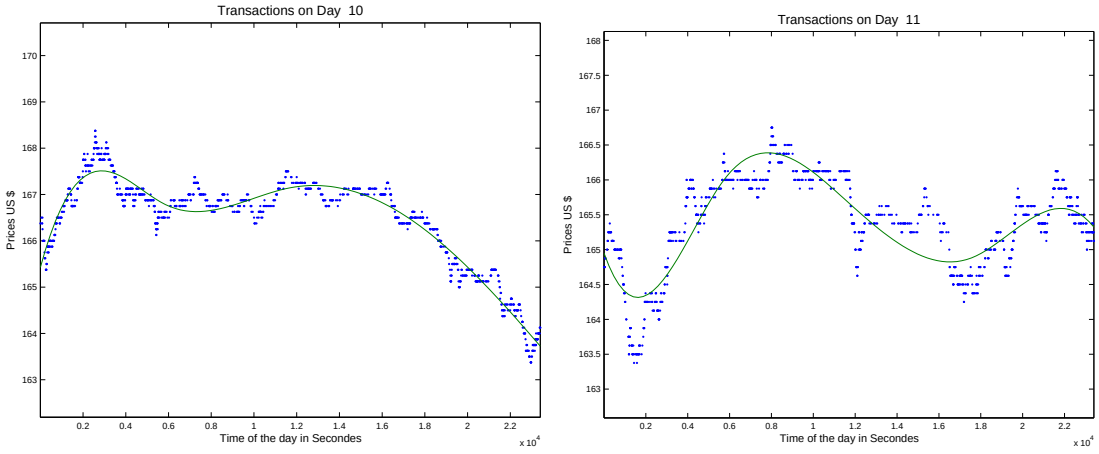

Figure 3: Two days of transactions for IBM (dashed curve),with smoothing splines (solid curve)

5.1 Prédiction

We forecast the futur transactions splines for three hours of day J between 10.30Hr. and 13.30 Hr. ($\triangle t_{out} = 3.0Hr$), from the past transactions (days $J-2$ and $J-1$ and half an

DSI-05, Decision Sciences Institute International Conference
Barcelona (Spain), 3-6 July 2005, pp. 159-169.

hour of day 9 between 10.00 Hr. and 10.30 Hr.), in this case ($\triangle t_{in} = 11.30Hr$). We have eliminated the transactions at the opening and closing of the NYSE, which are "outliers" without any correlation with the next hours. On Fig. 4 we can see four out-of-sample forecasting days superposed with the observations and smoothing splines (not known by the model). There is a good correlation between the out-of-sample forecasting and the observations.

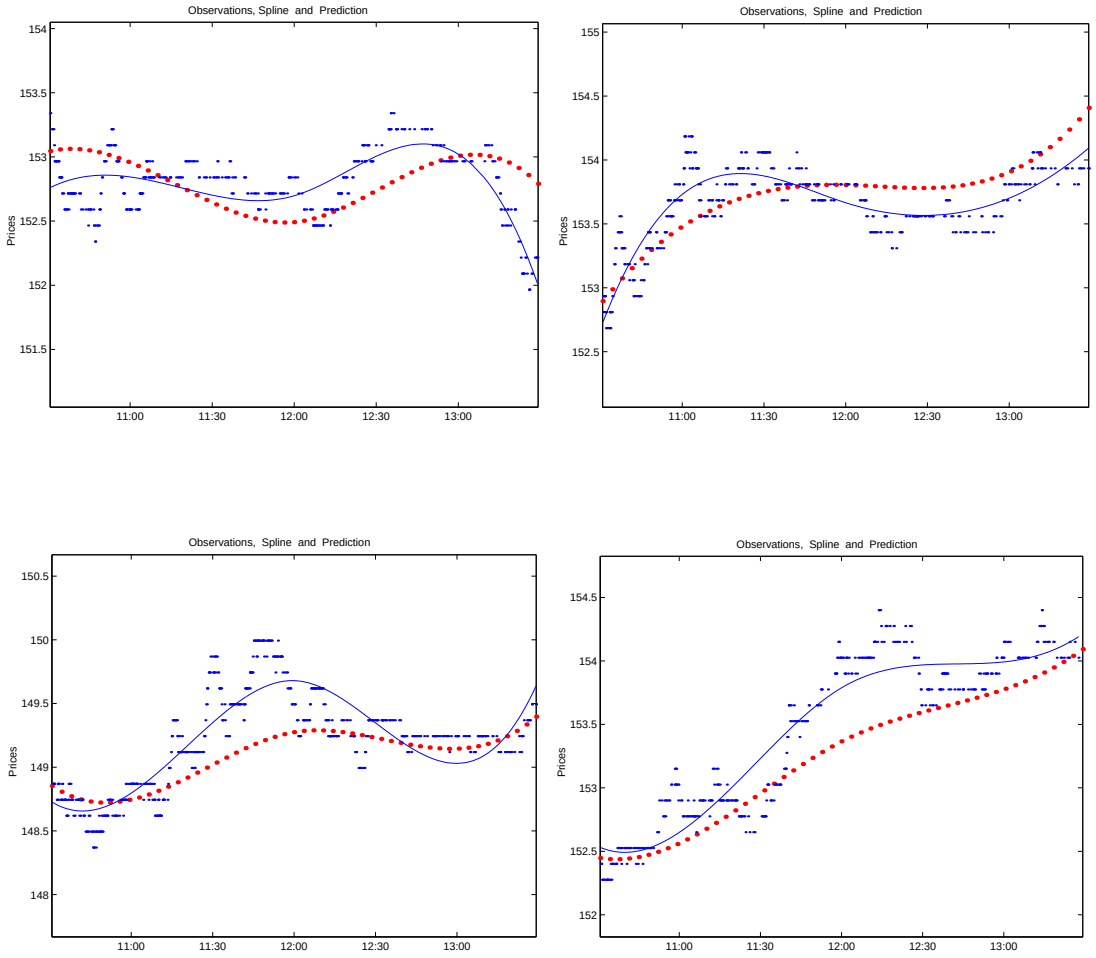

Figure 4: Four forecasting days for IBM stock price. Observations (Points); Smoothing splines (solid curve); Out-of-sample forecasting by the model (dashed curve)

6 Conclusion

We have presented a functional method for the clustering, modelling and forecasting of time series by functional analysis and neural networks. This method can be applied to all types of time series but is particularly effective when the observations are sparse, irregularly spaced, occur at different time points for each curve, or when only fragments of the curves are observed; standard methods completely fail in these circumstances. By the functional clustering, we can also realize the forecasting of multiple dynamic processes.

Data Mining III, Computational Sciences Institute International Conference
Barcelona (Spain), 3-6 July 2005, pp. 159-169.

[1] Y. Ait-Sahalia and P.A. Myland,(March, 2003),"The effects of random and discrete sampling when estimating continuous-time diffusions",*Econometrica*, Vol. 71, pp. 483-549.

[2] N. Benoudjit, M. Verleysen, "On the kernel widths in Radial-Basis Function Networks", Neural Processing Letters, Kluwer academic pub., vol. 18, no. 2, pp. 139-154, October 2003.

[3] G. Box and G.M. Jenkins,(1970), " Time series Forecasting and Control ", *Holden-Day* ,

[4] C. de Boor,(1978), " A Practical Guide to Splines ", *New York : Springer* ,

[5] A.P. Dempster, N.M. Laird, and D.B. Rubin,(1977), "Maximum likelihood from incomplete data via the EM algorithm ", *Jounal of the Royal Statistical Society, Ser. B* , Vol. 39, pp. 1-22.

[6] F. Diebold and J.-H. Lee and G.Weinbach, "Regime switching with time-varying transition probabilities, in non stationary time series analysis and cointegration", ed. by C. Hargreaves, 1994, pp 283-302, *Oxford University Press.*

[7] R.F. Engel, "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation",*Econometrica*, Vol. 50, pp. 987-1007, 1982.

[8] E.F. Fama, "Efficient Capital Markets : II",*The Journal of Finance*, Vol. XLVI, N° 5, pp. 1515-1617, 1991.

[9] J.D. Hamilton,(1994), " Time Series Analysis ", *Princeton University Press* ,

[10] S. Haykin,(1999), " Neural Networks - A comprehensive foundation ", *Prentice Hall*,

[11] J.A. Rice and C.O. Wu,(March 2001), "Nonpoarametric Mixed Effects models for Unequally Sampled Noisy Curves",*Biometrics*, Vol. 57, pp. 253-259.